NRG Technologies
T.: +994 (50) 2001335
E-mail: to@nrg.az

MAGNET
Payment Gateway

Merchant API External Specification
(API version 1. Spec. version 1.14)

Contents

0. Version history

Ver.	Date	Author	Comment
0.01	25.08.2017	Fariz Kazimov	Initial document version
1.01	02.11.2017	Fariz Kazimov	Final document version
1.02	07.11.2017	Vasif Aghayev	Fine tuning
1.03	04.01.2018	Vasif Aghayev	Template resources (7.3) added
1.04	05.01.2018	Vasif Aghayev	Template recommendations (7.4) added
1.05	05.01.2018	Fariz Kazimov	card.tpl template parameters changed in 7.5
1.06	05.01.2018	Fariz Kazimov	Changed datetime format (from hh to HH)
1.07	11.01.2018	Fariz Kazimov	{issuer} variable added to success.tpl
1.08	08.08.2018	Vasif Aghayev	New redirect.tpl template added (7.9)
1.09	08.08.2018	Vasif Aghayev	Status 22 changed to unsuccessful in 8.3
1.10	08.08.2018	Vasif Aghayev	Status classification added/changed in 8.3
1.11	10.10.2019	Fariz Kazimov	Card 'alias' parameter changed to 'token'
1.12	07.11.2019	Fariz Kazimov	Added explicit card data parameters in 5.3
1.13	12.11.2019	Fariz Kazimov	Amount parameter added in 5.4, 5.5, 5.6
1.14	05.12.2019	Fariz Kazimov	'09' status code added to 8.1

1. Document purpose

The current specification contains general provisions for integration of Merchant software for the purpose of interaction between Merchant and MAGNET E-Commerce System, as well as a detailed description of data structures, web service methods and associated XML/JSON messages.

2. General

Interaction between Merchant software and MAGNET E-Commerce System is performed by utilizing of RESTful Web Services. For security purposes the data transfer between client and server is encrypted by SSL/TLS technology.

Error codes and associated descriptions retuned by service are described on the last page of the document.

3. Connection

In case of VPN connection, it's needed to write the following entry in the hosts file of the operating system (or local DNS server) in order to match server certificate's Common Name (CN) attribute:

```
10.129.252.191  sandbox.api.pay.yigim.az
```

4. Authentication

The Merchant authentication is performed by HTTP Headers containing Merchant name and security signature.

```
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
```

The security signature is produced by concatenation of the parameter values of GET request in the order they appear in 'Accepted parameters' tables (if the value of optional parameter is missing then it is ignored) and the "secret key" assigned to Merchant. Concatenation of the parameter values with the "secret key" (UTF-8 encoded) should be hashed using digital signature algorithm and it’s bytes should be encoded by Base64:

```
BASE64(BYTE(MD5(VALUE1 + VALUE2 + ... + VALUEN+KEY)));
```

The Merchant name and Secret key are provided by acquirer.

Warning

Secret key is a confidential information that ensures the security of payments. If you suspect that this information has become available to third parties, immediately change the key through the Merchant Workplace.

5. Service commands

5.1 Payment initialization

URL: `https://.../payment/create`

Method: `GET`

This command is used for initial payment registration and optionally may start card linking procedure. Returns URL to redirect cardholder to for entering of card information.
Both SMS and DMS payment schemes are supported and determined by 'type' parameter. To retrieve payment status refer to section 5.2.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID for future usage
type	N	SMS - Single (default), DMS - Dual Message System
token	N	Optional card unique token (optional)
save	N	Save card data? y = yes, n = no
amount	Y	Payment amount in coins, i.e. 50.25 = 5025
currency	Y	Numeric ISO4712 currency code
biller	Y	Biller name provided by acquirer
description	N	Payment description for displaying on card input page
template	Y	Template name for card input page, see section 7
language	Y	Template language (2 letters: az/en/ru/etc.)
callback	N	Webhook URL triggered when payment status changes
extra	N	URL-encoded parameters: name1=value1;name2=value2

Optional 'callback' parameter must contain fully qualified URL (HTTP or HTTPS with valid SSL certificate) of merchant's payment status handler. When payment is completed this URL will receive a HTTP GET request with 'reference' parameter reflecting merchant defined unique payment (order) ID used for payment initialization:

```
GET <callback URL>&reference=REF0001
```

Then the concrete payment status may be obtained by calling appropriate API command (see section 5.2 for more details).

Output parameters:

Name	Description
url	Card input page URL to redirect cardholder
code	Response code, see section 8.2 for details
message	Response message

Sample GET request:

```
GET /payment/create
? reference=REF0001
& type=SMS
& token=CRD0001
& save=y
& amount=1
& currency=944
& biller=BLR0001
& description=Test payment
& template=TPL0001
& language=az
& callback=https://merchant.az/callback.jsp HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
X-Type: XML
```

Sample XML response:

```
<response>
  <url>https://.../payment/45617812104414</url>
  <code>0</code>
  <message>OK</code>
</response>
```

Sample JSON response:

```
{
  "response": {
    "url": "https://.../payment/45617812104414",
    "code": 0,
    "message": "OK"
  }
}
```

5.2 Payment status

URL: `https://.../payment/status`

Method: `GET`

This command is used to check payment status and get related information if applicable. Some of output parameters may be missing depending on payment status.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID

Output parameters:

Name	Description
reference	Unique payment (order) ID
datetime	Payment date and time (yyyy-MM-dd'T'HH:mm:ss.SSS)
type	SMS - Single, DMS - Dual Message System
token	Card unique token (if available)
pan	Masked card number (if available)
expiry	Card expiry date, format: mmyy (if available)
amount	Payment amount
currency	Numeric ISO4712 currency code
biller	Biller name provided by acquirer
system	Card payment system: Visa, MasterCard, etc.
issuer	Card issuing bank name (if available)
rrn	Unique acquirer transaction id
approval	Issuer authorization approval code
3ds	3-D Secure status (if supported), see section 8.3 for details
status	Payment status, see section 8.1 for details
code	Response code, see section 8.2 for details
message	Response message
extra	Extra parameters (may be configured by acquirer)

Sample GET request:

```
GET /payment/status?reference=REF0001 HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
X-Type: XML
```

Sample XML response:

```
<response>
  <reference>REF0001</reference>
  <datetime>2017-09-10T10:00:00.100</datetime>
  <type>SMS</type>
  <token>CRD0001</token>
  <pan>510068******0966</pan>
  <expiry>0920</expiry>
  <amount>1</amount>
  <currency>944</currency>
  <biller>BLR0001</biller>
  <system>MasterCard</system>
  <issuer>Chase Bank</issuer>
  <rrn>712345678901</rrn>
  <approval>016355</approval>
  <status>00</status>
  <code>0</code>
  <message>OK</code>
</response>
```

Sample JSON response:


```
{
  "response": {
    "reference": "REF0001",
    "datetime": "2017-09-10T10:00:00.100",
    "type": "SMS",
    "token": "CRD0001",
    "pan": "510068******0966",
    "expiry": "0920",
    "amount": "1",
    "currency": "944",
    "biller": "BLR0001",
    "system": "MasterCard", "issuer": "Chase Bank",
    "rrn": "712345678901",
    "approval": "016355",
    "status": "00",
    "code": 0,
    "message": "OK"
  }
}
```


5.3 Direct payment

URL: `https://.../payment/execute`

Method: `GET`

This command is used to perform direct payment with saved (tokenized) or explicit (card number, expiry date and card verification value) card data without redirecting cardholder to the card input page (unlike described in section 5.1).

Supplement of the explicit card data only allowed for merchants with a valid PCI DSS certificate.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID for future usage
type	N	SMS - Single (default), DMS - Dual Message System
token	C	Card unique token (optional)
pan	C	Full card number
expiry	C	Card expiry date (MMYY)
csc	C	Card verification value (CVV/CVV2/CVC)
amount	Y	Payment amount in coins, i.e. 50.25 = 5025
currency	Y	Numeric ISO4712 currency code
biller	Y	Biller name provided by acquirer
extra	N	URL-encoded parameters: name1=value1;name2=value2

Output parameters:

Name	Description
reference	Unique payment (order) ID
datetime	Payment date and time (yyyy-MM-dd'T'HH:mm:ss.SSS)
type	SMS - Single, DMS - Dual Message System
pan	Masked card number (if available)
amount	Payment amount
currency	Numeric ISO4712 currency code
biller	Biller name provided by acquirer
system	Card payment system: Visa, MasterCard, etc.
issuer	Card issuing bank name (if available)
rrn	Unique acquirer transaction id
approval	Issuer authorization approval code
3ds	3-D Secure status (if supported), see section 8.3 for details
status	Payment status, see section 8.1 for details
code	Response code, see section 8.2 for details
message	Response message

Sample request:

```
GET /payment/execute
? reference=REF0001
& type=SMS
& token=CRD0001
& amount=1
& currency=944
& biller=BLR0001
& description=Test payment HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
X-Type: XML
```

Sample XML response:

```
<response>
  <reference>REF0001</reference>
  <datetime>2017-09-10T10:00:00.100</datetime>
  <type>SMS</type>
  <pan>510068******0966</pan>
  <amount>1</amount>
  <currency>944</currency>
  <biller>BLR0001</biller>
  <system>MasterCard</system>
  <issuer>Chase Bank</issuer>
  <rrn>712345678901</rrn>
  <approval>016355</approval>
  <status>00</status>
  <code>0</code>
  <message>OK</code>
</response>
```

Sample JSON response:

```
{
  "response": {
    "reference": "REF0001",
    "datetime": "2017-09-10T10:00:00.100",
    "type": "SMS",
    "pan": "510068******0966",
    "amount": "1",
    "currency": "944",
    "biller": "BLR0001",
    "system": "MasterCard", "issuer": "Chase Bank",
    "rrn": "712345678901", "approval": "016355",
    "status": "00",
    "code": 0,
    "message": "OK"
  }
}
```

5.4 Payment clearance (capture)

URL: `https://.../payment/charge`

Method: `GET`

This command is used for DMS payments (type=DMS) which will clear (commit) the locked amount from the card. To retrieve appropriate payment information refer to section 5.2.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID
amount	N	Optional amount to be captured (must be <= of original)

Output parameters:

Name	Description
code	Response code, see section 8.2 for details
message	Response message

Sample request:

```
GET /payment/charge?reference=REF0001 HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
X-Type: XML
```

Sample XML response:

```
<response>
  <code>0</code>
  <message>OK</code>
</response>
```

Sample JSON response:

```
{
  "response": {
    "code": 0,
    "message": "OK"
  }
}
```

5.5 Payment cancellation (reverse)

URL: `https://.../payment/cancel`

Method: `GET`

This command is used for payment cancellation. The blocked amount is returned to the card instantly until settlement is done (before the end of the trading day), otherwise error will be returned. After settlement is over payment refund (see section 5.6) must be used.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID
amount	N	Optional amount to be reversed (must be <= of original)

Output parameters:

Name	Description
code	Response code, see section 8.2 for details
message	Response message

Sample request:

```
GET /payment/cancel?reference=REF0001 HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
```

Sample XML response:

```
<response>
  <code>0</code>
  <message>OK</code>
</response>
```

Sample JSON response:

```
{
  "response": {
    "code": 0,
    "message": "OK"
  }
}
```

5.6 Payment refund

URL: `https://.../payment/refund`

Method: `GET`

This command is used for payment refund. The cleared settlement amount is returned to the card after next clearing procedure with issuer bank.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID
amount	N	Optional amount to be refunded (must be <= of original)

Output parameters:

Name	Description
code	Response code, see section 8.2 for details
message	Response message

Sample request:

```
GET /payment/refund?reference=REF0001 HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
```

Sample XML response:

```
<response>
  <code>0</code><message>OK</code>
</response>
```

Sample JSON response:

```
{
  "response": {
    "code": 0,
    "message": "OK"
  }
}
```

5.7 Closing of batch

URL: `https://.../merchant/batch/close`

Method: `GET`

This command is used for closing of the trading day. All transactions made after last successfully closed batch will be sent to settlement ad payment cancellation will no longer take effect for them.

Accepted parameters:

Name	Required	Description
X-Merchant	Y	HTTP Header for merchant name
X-Signature	Y	HTTP Header for signature
X-Type	N	HTTP Header for response type, values: "XML" or "JSON"
reference	Y	Unique payment (order) ID

Output parameters:

Name	Description
code	Response code, see section 8.2 for details
message	Response message

Sample request:

```
GET /merchant/batch/close?reference=REF0001 HTTP/1.1
X-Merchant: Merchant
X-Signature: OaDZaBl6b13xIP+U9nulYQ==
X-Type: XML
```

Sample XML response:

```
<response>
  <code>0</code>
  <message>OK</code>
</response>
```

Sample JSON response:

```
{
  "response": {
    "code": 0,
    "message": "OK"
  }
}
```

6. P2P transfers

The functionality of P2P transfers (transfer from card to card) allows you to transfer money from one card of international payment systems (Visa and MasterCard) to another.

P2P transfers are the same payments with a small difference. Depending on the combination of sender and recipient card type (saved card vs. not saved card) the additional API request parameter (extra parameter) or template form parameter applies (for P2P template):

Sender	Recipient	Additional parameter
Saved card	Saved card	API: recipient=<token>
Saved card	Not saved card	Template: <input type="text" name="recipient"/>
Not saved card	Saved card	API: recipient=<token>
Not saved card	Not saved card	Template: <input type="text" name="recipient"/>

7. Merchant templates

7.1 General information

MAGNET has support for fully customizable and localizable merchant templates for card information input (standard payment and P2P). There are 4 types of payment templates available:

- Card information input template (card.tpl);
- Surcharge notification and confirmation template (surcharge.tpl);
- 3-D Secure redirect template (redirect.tpl);
- Failed payment template (failure.tpl);
- Successful payment template (success.tpl).

Template name format: `TPLXXXX`, for example: `TPL0001`, `TPL0002`, etc.

7.2 Localization

Template localization is achieved by creating a separate file for each language placed into appropriate language directory in the following format:

`TPL0001/az/card.tpl` - for template in Azeri
`TPL0001/en/card.tpl` - for template in English
`TPL0001/ru/card.tpl`- for template in Russian

7.3 Template resources

All static template resources (images, scripts, styles, etc.) reside in the following folder:

`/assets/<merchant name>/<template name>/`

In order to use them inside templates follow the sample guidelines below:

```
<link href="/assets/MRC0001/TPL0001/css/style.css" .../>
<script src="/assets/MRC0001/TPL0001/js/jquery.js" ...></script>
<img src="/assets/MRC0001/TPL0001/images/logo.png" .../>
```

7.4 Template recommendations

- For security reasons use autocomplete="off" attribute for all forms;
- Don't use external (resources scripts, images, etc.), or use only ones with https:// scheme.

7.5 Card information input template (card.tpl)

Template variables:

Name	Description
{url}	Action value for POST form
{id}	Internal payment id
{reference}	Unique payment (order) ID
{amount}	Payment amount (double)
{currency}	Payment currency code (3 letters, ISO4217)
{description}	Payment description
{extra:name}	Extra parameter value, replace name with parameter name

Form POST parameters:

Name	Required	Description
id	Y	Hidden parameter obtained from {id} template variable
pan	Y	Card number (16-19 digits)
month	Y	Card expiry month (date format: mm)
year	Y	Card expiry year (date format: yy)
holder	Y	Cardholder name
csc	N	Card security code (CVV, CVV2, CVC, etc.)
recipient	N	Recipient card number for P2P payments (16-19 digits)

7.6 Surcharge notification and confirmation template (surcharge.tpl)

In case when payment has a surcharge then this template will be displayed for cardholder confirmation before card authorization. Sending the POST request with hidden {id} value will proceed with card authorization.

Template variables:

Name	Description
{url}	POST HTML form's action value
{id}	Internal payment id
{surcharge}	Payment surcharge (double)
{currency}	Payment currency code (3 letters, ISO4217)
{description}	Surcharge description (if available)

Form POST parameters:

Name	Required	Description
id	Y	Parameter obtained from {id} template variable

7.7 Failed payment template (failure.tpl)

Template variables:

Name	Description
{url}	URL for redirecting the cardholder back to merchant
{id}	Internal payment id
{surcharge}	Payment surcharge (double)
{currency}	Payment currency code (3 letters, ISO4217)
{description}	Surcharge description (if available)

Form POST parameters:

Name	Required	Description
id	Y	Hidden parameter obtained from {id} template variable

7.8 Successful payment template (success.tpl)

Template variables:

Name	Description
{url}	URL for redirecting the cardholder back to merchant
{id}	Internal payment id
{reference}	Unique payment (order) ID
{datetime}	Payment date and time (yyyy-MM-dd'T'HH:mm:ss.SSS)
{pan}	Masked card number
{expiry}	Card expiry date, format: mmyy
{amount}	Payment amount (double)
{surcharge}	Payment surcharge (double)
{currency}	Payment currency code (3 letters, ISO4217)
{issuer}	Card issuer name (if available)
{description}	Surcharge description (if available)
{rrn}	Unique acquirer transaction id
{approval}	Issuer authorization approval code
{extra:name}	Extra parameter value, replace name with parameter name

7.9 3-D Secure redirect template (redirect.tpl)

In case when entered card information is a subject for 3-D Secure verification, this template will be presented to cardholder informing him about redirection to the card's issuer domain. This template may contain information about upcoming redirect with timeout and automatic redirect and a button for manual redirection as well as a blank page with immediate automatic POST form to the ACS URL.

Template variables:

Name	Description
{url}	ACS URL to make POST request to
{pareq}	Payer Authentication Request (PaReq) value
{returnURL}	Return URL to redirect cardholder back after verification

Form POST parameters:

Name	Required	Description
PaReq	Y	Payer Authentication Request (PaReq)
TermUrl	Y	Return URL to redirect cardholder back after verification

8. References

8.1 Authorization status codes

Response codes are supplied by the merchant's bank to payment gateway provider after making contact with the customer's card issuer.

Common types of responses from the banks include:

- Transaction approved: Your customer's transaction was processed successfully.
- Insufficient funds: The issuer bank noted that there is not enough money available on the card.
- Card expired
- Refer to issuer: This can mean many different things, ultimately you will need to tell your customer to contact their card issuing bank.

Please find below a more complete list of the status codes you may receive.

code - Status is not yet determined;
code - Approved transaction;
code - Failed transaction.

Code	Description
S0	Newly created transaction, waiting for card data input
S1	Pre-authorized DMS transaction (call 'charge' or 'cancel' within 30 days)
S2	Transaction is in progress
S3	Unknown error
S4	Reversed transaction (cancelled)
S5	Refunded transaction
S7	System malfunction
00	Approved
01	Decline, refer to issuer
02	Decline, expired card
03	Decline, invalid amount
04	Decline, inactive card
05	Decline, insufficient funds
06	Decline, suspected fraud
07	Decline, exceeds withdrawal limit
08	Format error
09	Issuer timeout (merchant should call 'cancel' transaction by themself)

8.2 Response error codes

Response codes indicate the status of request processing and must be checked first. Generally it is used for input parameters validation, such as format validation or signature validation.

code - No error;
code - Error.

Code	Description
0	OK (No error)
1	Invalid merchant name specified (HTTP Header 'X-Merchant' value)
2	Invalid signature specified
3	Access denied (Merchant's IP is not in access list)
4	Not permitted (method is not allowed for Merchant)
5	Invalid parameter [name] specified
6	System error

8.3 3-D Secure status codes

3-D Secure status code can have one of the following values.

code - Successful 3-D Secure authentication;
code - Non-participation and attempts;
code - Failed 3-D Secure authentication, transaction will not proceed.

Code	Description
00	Successful 3-D Secure authentication
10	Card is not in 3-D secure card range defined by issuer
20	Failed 3-D Secure authorization
21	Cardholder is not a member of 3D Secure scheme
22	Cardholder 3-D secure authorization is unavailable
23	Error message received from ACS server
24	3-D secure authorization ended with system error
25	3-D secure authorization was attempted by wrong card scheme
30	Cardholder 3-D secure authorization using attempts ACS server